CLAIM AMENDMENTS

Please amend the claims as follows:

53. (currently amended) A method comprising:

indicating in a configuration register that a vector buffer is in use, wherein said vector buffer is one of a plurality of vector buffers comprising a vector buffer pool; and

transferring vector data between memory and a register file via said vector buffer, wherein said configuration register, said register file and said vector buffer pool are local to a processor, and said transferring is initiated by a first program;

generating a vector transfer unit exception if a second program attempts to transfer vector data between memory and a register of the register file via any vector buffer of the vector buffer pool, if said configuration register indicates that said vector buffer is in use.

54. (previously added) The method of claim 53, further comprising:

indicating in the configuration register that the vector buffer is not in use.

55. (cancelled)

56. (currently amended) The method of claim ~~55~~ 53, further comprising:

indicating in said configuration register that said vector buffer pool is locked if said second program attempts said transfer.

57. (previously added) The method of claim 56, wherein said indicating in said configuration register that said vector buffer pool is locked comprises:

setting a vector buffer lock bit in said configuration register.

58. (previously added) The method of claim 57, further comprising:

clearing said vector buffer lock bit in said configuration register; and

performing said transfer by said second program.

59. (currently amended) The method of claim 53, wherein said indicating in the configuration register that the vector buffer is ~~not~~ in use comprises:

issuing a free vector buffer instruction.

60. (previously added) The method of claim 58, wherein said indicating in the configuration register that the vector buffer is not in use further comprises:

clearing a vector buffer in-use bit in said configuration register.

61. (previously added) The method of claim 53, wherein said transferring comprises:

transferring said vector data from said memory to said vector buffer as a result of executing a load vector instruction.

62. (previously added) The method of claim 53, wherein said transferring comprises:

transferring said vector data from said vector buffer to said register file as a result of executing a move vector from buffer instruction.

63. (previously added) The method of claim 53, wherein said transferring comprises:

transferring said vector data from said register file to said vector buffer as a result of executing a move vector to buffer instruction.

64. (previously added) The method of claim 53, wherein said transferring comprises:

transferring said vector data from said vector buffer to said memory as a result of executing a store vector instruction.

65. (previously added) The method of claim 53, wherein said configuration register, said register file and said vector buffer pool are on the same substrate as the processor.

66. (currently amended) A system comprising:

a memory;

a data processor coupled to said memory, said data processor comprising:

a data cache;

a register file;

a configuration register;

a vector transfer execution unit; and

a vector buffer pool including a plurality of vector buffers, said vector transfer execution unit configured to transfer vector data between said memory and said register file via said vector buffer pool;

wherein said vector transfer execution unit is further configured to indicate, in said configuration register, whether or not a vector buffer of said vector buffer pool is in use by a first program;

wherein said vector transfer execution unit is further configured to prevent a second program from transferring data between said memory and said register file via said vector buffer pool if said vector buffer pool is in use.

67. (cancelled)

68. (cancelled)

69. (currently amended) The system of claim ~~68~~ 66, wherein said vector transfer execution unit comprises a vector transfer instruction queue configured to store a plurality of vector transfer instructions to transfer vector data between said memory and said register file via said vector buffer pool.

70. (previously added) The system of claim 66, wherein said vector transfer execution unit is further configured to transfer said vector data from said memory to said vector buffer as a result of executing a load vector instruction.

71. (previously added) The system of claim 66, wherein said vector transfer execution unit is further configured to transfer said vector data from said vector buffer to said register file as a result of executing a move vector from buffer instruction.

72. (previously added) The system of claim 66, wherein said vector transfer execution unit is further configured to transfer said vector data from said register file to said vector buffer as a result of executing a move vector to buffer instruction.

73. (previously added) The system of claim 66, wherein said vector transfer execution unit is further configured to transfer said vector data from said vector buffer register to said memory as a result of executing a store vector instruction.

74. (previously added) The system of claim 66 wherein said configuration register comprises:

a length of said vector buffers;

a number of vector buffers in said vector buffer pool; and

an identification of an active vector buffer.

75. (previously added) The system of claim 66 wherein said configuration register comprises:

a code for an exception caused by a vector transfer unit instruction.

76. (previously added) The system of claim 66 wherein said configuration register comprises:

a vector buffer pool in use bit to indicate whether said vector buffer pool is available or in use; and

a vector buffer pool lock bit to indicate whether said vector buffer pool is allocated to a program.

77. (currently amended) A computer-readable medium having stored thereon instructions which, when executed by a processor, configure the processor to:

indicate in a configuration register that a vector buffer is in use, wherein said vector buffer is one of a plurality of vector buffers comprising a vector buffer pool; and

transfer vector data between memory and a register file via said vector buffer, wherein

said configuration register, said register file and said vector buffer pool are local to a processor, and said transferring is initiated by a first program;

generate a vector transfer unit exception if a second program attempts to transfer vector data between memory and a register via any vector buffer of the vector buffer pool, if said configuration register indicates that said vector buffer is in use.

78. (previously added) The computer-readable medium of claim 77 which further configures the processor to:

indicate in the configuration register that the vector buffer is not in use.

79. (cancelled)

80. (currently amended) The computer-readable medium of claim ~~79~~ 77 which further configures the processor to:

indicate in said configuration register that said vector buffer pool is locked if said second program attempts said transfer.

81. (previously added) The computer-readable medium of claim 80 which further configures the processor to:

set a vector buffer lock bit in said configuration register.

82. (previously added) The computer-readable medium of claim 81 which further configures the processor to:

clear said vector buffer lock bit in said configuration register; and

perform said transfer by said second program.

83. (previously added) The computer-readable medium of claim 77 which further configures the processor to:

issuing a free vector buffer instruction.

84. (previously added) The method of claim 82, wherein said indicating in the configuration register that the vector buffer is not in use further comprises:

clear a vector buffer in-use bit in said configuration register.

85. (previously added) The computer-readable medium of claim 77 which further configures the processor to:

transfer said vector data from said memory to said vector buffer as a result of executing a load vector instruction.

86. (previously added) The computer-readable medium of claim 77 which further configures the processor to:

transfer said vector data from said vector buffer to said register file as a result of executing a move vector from buffer instruction.

87. (previously added) The computer-readable medium of claim 77 which further configures the processor to:

transfer said vector data from said register file to said vector buffer as a result of executing a move vector to buffer instruction.

88. (previously added) The computer-readable medium of claim 77 which further configures the processor to:

transfer said vector data from said vector buffer to said memory as a result of executing a store vector instruction.